

# X - Sistemi datoteka

## SADRŽAJ

**10.1** Pojam sistema datoteka

**10.2** Pojam datoteke

**10.3** Metodi pristupa datotekama

**10.4** Zaštita i prava pristupa

**10.5** Pojam direktorijuma

**10.6** Dodela prostora za datoteke

**10.7** Realizacija sistema datoteka

**10.8** Strukture podataka za pamćenje

**10.9** Vrste sistema datoteka

# 10.1 - Pojam sistema datoteka

- Podaci se **uvek menjaju u radnom području** primarne memorije, a zatim se tako promenjeni **prenose u neku strukturu na sekundarnu memoriju**
  - U toku izvršavanja programa **podaci su uvek u radnoj, primar.memoriji**
  - U sekundarnoj memoriji oni se čuvaju u blokovima (klasterima) u obliku: **datoteka, baze podataka, registara ili internih struktura.**
  - Podaci u aplikaciji **moraju efikasno** da koriste sekundarnu memoriju
  - Baze podataka po pravilu **zauzimaju znatno više mem. prostora** nego sekvencijalne datoteke, naročito kada je reč o malom broju podataka
  - Upravljanje sadržajem datoteka direktno **je povezano sa premeštanjem blokova podataka** između primarne i sekundarne memorije
  - Podaci se prebacuju iz primarne u sekundarnu **radi njihovog trajnog čuvanja**, a iz sekundarne u primarnu memoriju **radi obrade**
  - Operativni sistem koristi **različite načine za organizaciju i kontrolu pristupa podacima** koji se nalaze na sekundarnoj memoriji
- Sistem datoteka (*File System*) je pojam koji podrazumeva logičke strukture podataka i softverske rutine koje se koriste za kontrolu pozivanja i smeštanja podataka na sekundarnu memoriju.**

# 10.1 – Pojam sistema datoteke

- U većini slučajeva, sistem datoteka je **nevidljiv aspekt operativnog sistema** koji obezbeđuje mehanizam **za čuvanje i pristup datotekama i programima** koji pripadaju operativnom sistemu ili korisnicima
- Zadatak sloja za upravljanje datotekama je **da obezbedi punu slobodu upravljanja podacima** koji se nalaze u datotekama.
- Ovakva sloboda podrazumeva:
  - ✓ **mogućnost proizvoljnog dodavanja** podataka u fajl,
  - ✓ **brisanje** podataka iz fajla,
  - ✓ **direktno pristupanje** svakom podatku u fajlu.
- Punu slobodu upravljanja podacima omogućuje **predstavljanje sadržaja datoteke kao niza bajtova**, pod uslovom da se taj niz može, po potrebi **produžavati ili skraćivati**, kao i da se bajtovima iz niza može **proizvoljno pristupati**
- Ovakva predstava datoteke dozvoljava da se iznad nje izgrade **različiti pogledi na sam fajl**.
- **Različiti operativni sistemi – različiti tipovi sistema datoteka**
- Sistem datoteka se sastoji od dva dela: ***datoteka i direktorijumima***

# 10.2 – Pojam datoteke

- Operativni sistem pruža **uniformni logički pogled** na sekundarnu memoriju **apstrakujući fizičke osobine** pojedinog memorijskog uređaja i definišući **logičku jedinicu – datoteku** (*file*).
- Datoteka predstavlja **fizički skup podataka** u sekundarnoj memoriji - tako je vidi OS, ali predstavlja i **imenovani skup informacija** koji čine **logičku celinu** (tekst, slika, muzika ili video) - tako je vidi korisnik
- Datoteke sadrže podatke i programe koji se **smeštaju na različite fizičke medijume** kao što su magnetne trake, diskovi, USB i slično.
- Informacije smeštene u datoteci **definiše kreator datoteke**.
- **Različiti tipovi informacija** mogu biti smešteni u datoteci: izvršni program, objektni program, numerički podaci, tekst, itd.
- Datoteka **ima određenu definisanu strukturu** saglasno njenoj upotrebi.
- Podaci jedne datoteke su **često smešteni na različitim segmentima diska**. Da bi korisnik mogao svaki put da ih vidi i dosegne putem definisanog imena, operativni sistem **mora posedovati niz kontrolnih podataka za svaku od datoteka – metapodaci ili atributi datoteke**.

# 10.2 - Pojam datoteke

Datoteka se sastoji od:

- 1. podataka** - smeštenih u sekundarnoj memoriji
  - 2. metapodataka** - definišu ime, tip, raspored blokova u memoriji, prava i vlasništvo, veličinu, vreme, datum - **kontrolni blok datoteke (FCB)**
- Mnogi operativni sistemi razlikuju **više tipova datoteka**:
- 1. regularne**: standardne korisničke datoteke (obični nizovi bajtova) koje mogu biti *ASCII* ili binarni fajlovi (ovu razliku pravi korisnik, a ne OS). *ASCII* fajlovi sadrže tekst, a binarni fajlovi bilo šta drugo.
  - 2. uređajne**: pod *Unix*-om uređajima pridružujemo fajlove (ako želimo nešto da stampamo to šaljemo-pišemo u odgovarajući fajl).
  - 3. specijalne**: zavise od OS-a. Na primer, pod *DOS*-om imamo specijalan fajl koji sadrži ime diska
  - 4. direktorijumi**: sistemski fajlovi koji sadrže informacije o strukturi sistema datoteka
- Datoteke se **prema nameni** dele na: **izvršne i datoteke sa podacima**.
- Prema **formatu podataka** datoteke mogu biti **tekstualne i binarne**

# 10.2 - Logička struktura datoteke

**1. Datoteka je niz bajtova** - najjednostavnija struktura gde OS ne vodi računa o tome šta se u njoj nalazi, **sve što vidi to je niz bajtova**. Korisnički programi su oni koji treba da dodeljuju značenje fajlovima (tj. da znaju kako treba da posmatraju/tretiraju taj niz bajtova). Ovu realizaciju koriste *DOS, UNIX, Windows*.

**2. Datoteka je niz slogova iste veličine** - predstavljaju strukture zapisa, pri čemu je jedan zapis može biti **fiksne ili promenljive dužine**. Primer zapisa može da bude jedan red u datoteci - **zastareli tip**.

**3. Datoteke u obliku B-stabla** - Predstavlja složenu strukturu gde su podaci strogo formatirani i kod kojih je **potrebno brzo pretraživanje** datoteke po zadatom ključu.

- **Pravila imenovanja fajlova razlikuje se od sistema do sistema.**
- Uglavnom svi sistemi **koriste niz znakova za imenovanje fajlova.**
- Neki sistemi dozvoljavaju i korišćenje **specijalnih znakova i brojeva.**
- **Neki prave razliku između malih i velikih slova-UNIX a neki ne-DOS**
- **DOS** - imena fajlova se sastoje iz 2 dela: **ime fajla(8).ekstenzija(3)**
- **UNIX** dozvoljava imena dužine do **255 karaktera.**

# 10.2 - Tipovi datoteke

- U većini OS, svakoj datoteci se pridružuje **tip** - ekstenzija imena
- Tip se određuje **u zavisnosti od aplikacije** u kojoj je izrađena i **formata i namene podataka** koji se čuvaju u datoteci.
- Informacija o tipu datoteke pomaže operativnom sistemu **da poveže datoteku sa aplikacijom** u kojoj se datoteka može koristiti.
- Neki od najpoznatijih tipova datoteka su:

<i>Tip datoteke</i>	<i>Vrsta datoteke</i>
Exe,com	Binearna izvršna datoteka
Obj	Prevedeni izvršni kod
Lib, dll	Statičke i dinamičke datoteke
Bat	Izvršna datoteka (paketna obrada)
Doc, txt	Tekstualne datoteke
Zip, rar, arj	Arhivirane datoteke
JPEG,bmp,PCX	Rasterske datoteke
MP3, WAW	Audio datoteke
MPEG, wmv	Multimedijalne datoteke

# 10.2 - Operacije sa datotekama

1. **Kreiranje datoteke** - Neophodna su dva koraka: određivanje prostora u fajl sistemu i unošenje odgovarajuće stavke u direktorijum.
2. **Upis u datoteku** - sistemski poziv gde se specificira i ime datoteke i podaci koji treba da se upišu u datoteku. Za dato ime datoteke sistem pretražuje direktorijum da bi pronašao lokaciju datoteke.
3. **Čitanje datoteke** - sistemski poziv specificira ime datoteke i mesto u memoriji gde sledeći očitani blok datoteke treba da se smesti. I ovde se pretražuje direktorijum, a odgovarajuća stavka treba da sadrži pointer na blok koji treba sledeći da se pročita. Kada se čitanje bloka obavi, taj pointer se ažurira.
4. **Pozicioniranje u datoteci** – ne predstavlja pravu U/I aktivnost, već se samo pretražuje direktorijum i u odgovarajućoj stavci se pointer tekuće pozicije resetuje da ukazuje na početak datoteke
5. **Brisanje datoteke** - Za brisanje datoteke se u direktorijumu traži specificirana datoteka, a onda se sav prostor dodeljen datoteci oslobađa, a dotična stavka u direktorijumu poništava (invalidira).



# 10.2 - Operacije sa datotekama

- Za sve opisane operacije **potrebno je pretraživanje direktorijuma**.
- Da bi olakšao pretraživanje OS **kreira malu tabelu sa informacijama** o svim trenutno otvorenim datotekama.
- Kada datoteka više nije potrebna, ona se **zatvara i odgovarajuća stavka se briše** iz tabele otvorenih datoteka.
- Većina sistema ipak zahteva **da se datoteke eksplicitno otvaraju** od strane programera - sistemski poziv (*open*) koji dodeljuje jedinstven ID
- Svaki OS ima **svoj skup sistemskih poziva** za rad sa datotekama:
  - **CREATE** (ime) - kreiranje nove, prazne datoteke
  - **DELETE** (ime) - brisanje datoteke
  - **OPEN** (ime) – otvaranje datoteke, vraća *file descriptor* – **fd**
  - **CLOSE** (fd) - zatvaranje datoteke koja je prethodno otvorena
  - **READ** (fd) – čita podatke iz otvorene datoteke od trenutne pozicije
  - **WRITE** (fd, podaci) – upisivanje podataka od trenutne pozicije
  - **APPEND**(fd, podaci)-isto kao WRITE, podaci se dodaju na kraj fajla
  - **SEEK** (fd, pos) - pomera pokazivač fajla na zadatu poziciju
  - **GETATTRIBUTES** (ime), **SETATTRIBUTES** (ime, atr)

# 10.3 - Metodi pristupa datotekama

**Sekvencijalni pristup** - podaci se prosleđuju **tačnim redosledom, jedan iza drugog**, u odnosu na vrednost tekućeg pokazivača

- Čitanje jednog bloka podataka automatski **uvećava vrednost pokazivača** (*current pointer, CP*), a slično se dešava i kod upisa.
- Moguće je vršiti **čitanje podataka samo u redosledu** u kome su upisani, odnosno **vršiti upis uzastopnih blokova podataka**
- Da bismo pročitali **n-ti bajt, moramo očitati prethodnih n-1 bajtova**
- Ovakav pristup zasnovan je na modelu datoteka koje su upisivane na **magnetne trake** koje su po svojoj prirodi sekvencijalni uređaji.
- Sekvencijalni pristup zahteva da **postoji mogućnost premotavanja datoteke na početak**, tako da vrednost pokazivača bude nula ( $CP=0$ ).
- Pri sekvencijalnom pristupu, mogu se izvesti **sledeće operacije**:
  - ✓ **čitanje** sledećeg bloka
  - ✓ **upis** u sledeći blok
  - ✓ **pozicioniranje na početak**

# 10.3 - Metodi pristupa datotekama

**Direktni pristup**-karakteristično za model datoteka upisanih na disk.

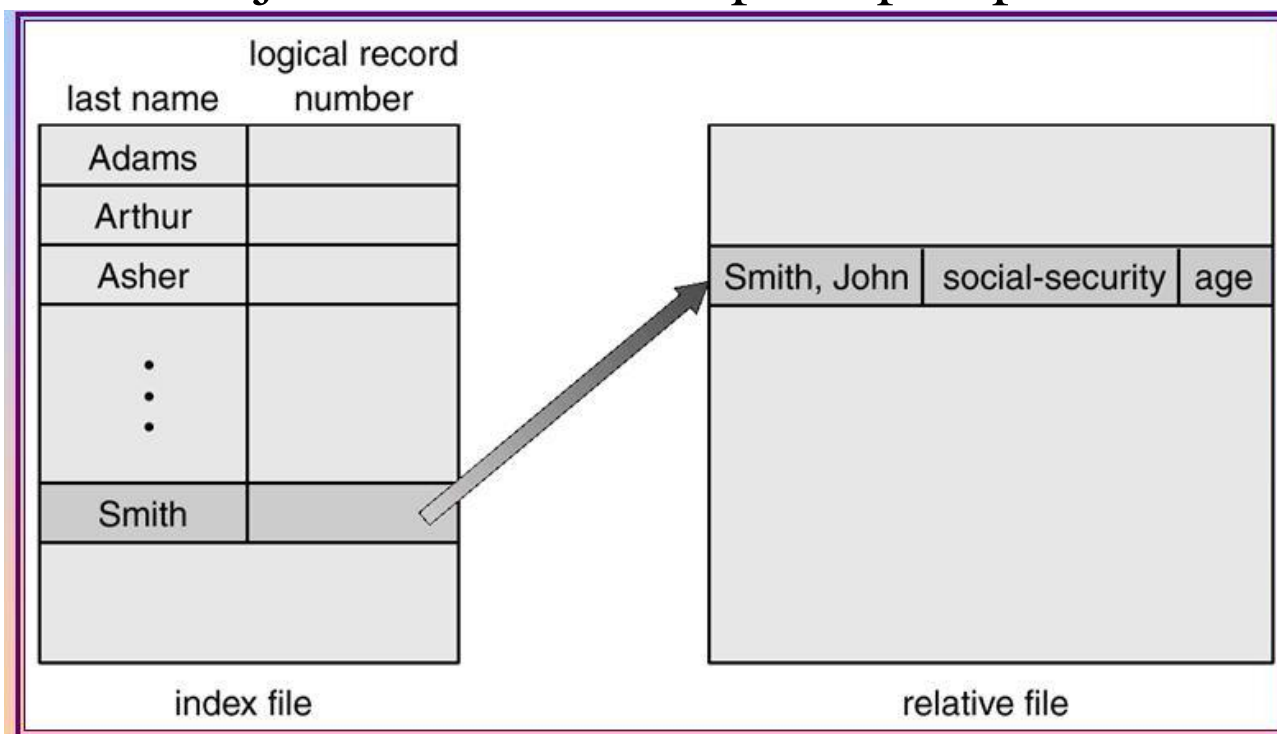
- Ovde se datoteka vidi kao numerisana sekvenca slogova fiksne dužine.
- Uz navođenje relativnog broja bloka može se pristupiti proizvoljnom slogu datoteke tj. možemo ažurirati bilo koji bajt unutar fajla
- Direktna metoda pristupa omogućava pristup bilo kom delu datoteke, tako što se odredi njegova pozicija na disku, a zatim pristupi podacima.
- Direktnan pristup omogućava korisniku da pristupi krajnjem bloku datoteke bez čitanja prethodnog sadržaja.
- Operacije koje se mogu izvesti su:
  - ✓ čitanje n-tog bloka
  - ✓ upis u n-ti blok
  - ✓ pozicioniranje na n-ti blok
  - ✓ čitanje sledećeg bloka
  - ✓ čitanje prethodnog bloka

# 10.3 - Metodi pristupa datotekama

**Indeksni pristup** – za svaku datoteku koristi se **dodatna indeksna datoteka** koja koristi ukazatelje na svaki od blokova datoteke.

- Svakoj datoteci pridružena je **indeksna datoteka**, uređena po nekom kriterijumu, pomoću koje se prilikom čitanja brzo može naći odgovarajući zapis.
- Da bi se pronašla stavka u datoteci, **najpre se pretražuje indeksna datoteka** i onda koristi ukazatelj da bi se direktno pristupilo podacima

- Prilikom upisa novog zapisa u datoteku, **ažurira se i indeksna datoteka**.
- Jako **brza metoda** za pronalaženje podataka pa se koristi kod **baza podataka**



# 10.4 - Zaštita i prava pristupa

- Kontrola pristupa i promene sadržaja datoteka je **jedna od veoma značajnih funkcija** koje operativni sistem mora da ispuni.
- Kod **Unix/Linux** sistem zaštita se ostvaruje pomoću dva mehanizma:
  1. **Definisanja vlasničke kategorije** – Vlasnik, Grupa i Svi
  2. **Definisanje pristupnih prava** - proizvoljna kombinacija osnovnih prava: Čitaj (*Read*), Piši (*Write*) i Izvršavaj (*Execute*)
- Kod **Windows** sistema, administracija kontrole pristupa i prava nad datotekama je ostvarena **kroz sistem Korisnika i Grupa**. Korisnik pripada **Grupi** koja može pripadati nadgrupi itd. Pri određivanju prava se poštuju principi: **nasleđivanja, sabiranja dozvola, jače zabrane**.
- Dozvole za rad sa određenim datotekama se formiraju **kao standardne dozvole u koje spadaju**: Čitanje, Pisanje, Puna kontrola, Modifikacija, Čitanje i Izvršavanje

# 10.5 – Pojam direktorijima

- Sadrži kontrolne blokove **datoteka** koje su u njemu logički smeštene.
- Svi **direktorijimi** sa svojim datotekama obrazuju sistem datoteka.
- Informacije koje se za svaku datoteku čuvaju u direktorijumu **variraju od jednog do drugog OS** i one se obično odnose na:
  - **Ime datoteke** - Simboličko ime datoteke.
  - **Tip datoteke** - Za one sisteme koji podržavaju različite tipove datoteka.
  - **Lokacija** - pointer na uređaj i lokaciju na tom uređaju gde je datoteka
  - **Veličina** - trenutna veličina datoteke (u bajtovima, rečima ili blokovima)
  - **Tekuća pozicija**-pointer na **tekuću poziciju za čitanje ili upis** u datoteku.
  - **Zaštita**-podatak o upravljanju pristupu radi čitanja, upisa, izvršavanja, dr.
  - **Brojač korišćenja**-ukazuje na broj procesa koji trenutno koriste datoteku
  - **Vreme, datum i identifikacija** procesa
- Operacije koje se mogu izvesti nad **direktorijumima** su:
  - ✓ **Prikazivanje** sadržaja direktorijuma – listanje
  - ✓ **Pretraživanje** direktorijuma
  - ✓ **Promena imena** direktorijuma
  - ✓ **Izrada i brisanje** direktorijuma

# 10.5 – Pojam direktorijima

- Direktorijum se sastoji od **skupa kontrolnih blokova svih datoteka** koje mu pripadaju i **skup meta podataka** koji opisuju njegovu strukturu.
- Da bi smo zapisali ove informacije za neku datoteku potrebno nam je **od 16 do 1000 bajtova**. Tako se direktorijum koji se nalazi na uređaju može učitavati u memoriju samo u manjim delovima, po potrebi
- Direktorijumi se u okviru sistema datoteka **moraju implementirati tako da obezbede efikasno pronalaženje** postojećih datoteka što se postiže tehnikama **B+ stabla i heš tabelama**.

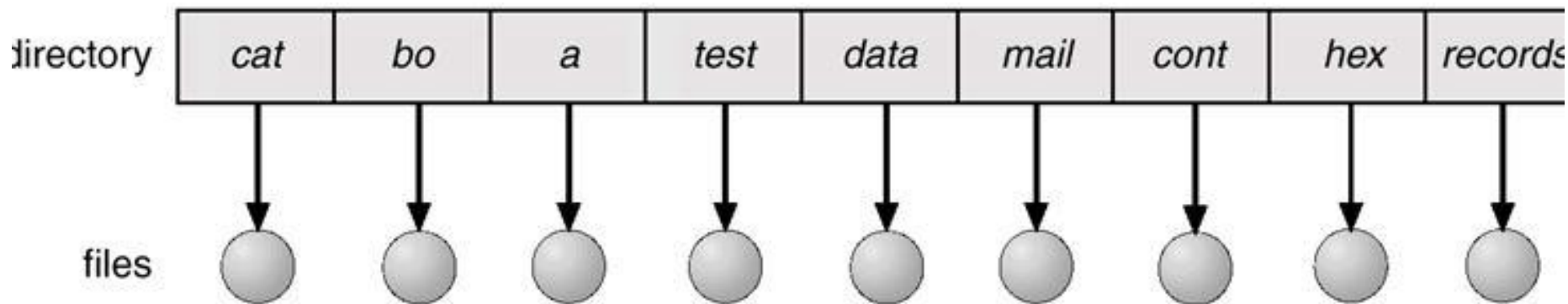
Logička struktura direktorijuma se može **organizovati na više načina**:

1. na jednom nivou,
2. na dva nivoa
3. na više nivoa (stablo direktorijuma)

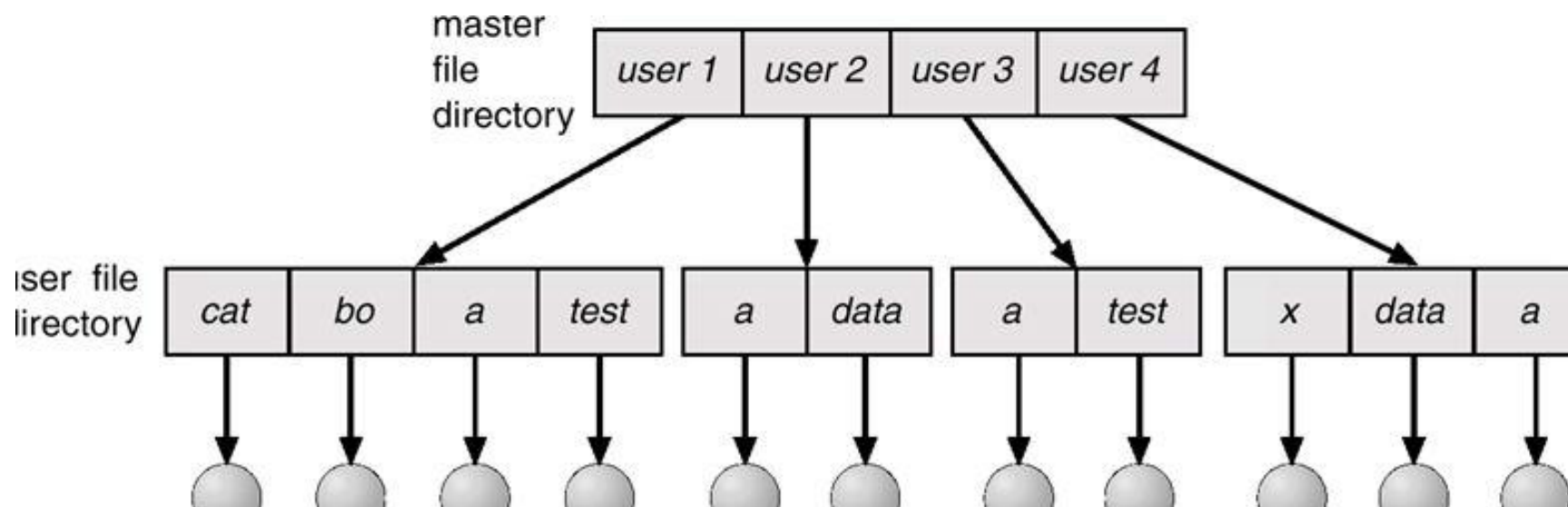
Koriste se sledeće strukture kod realizacije:

- 1. linearna lista** - zahteva linearno pretraživanje za pronalaženje odgovarajuće stavke. Jednostavan ali troši previše vremena.
- 2. heš tabela**. Najveći problem sa heš tabelama jeste njihova generalno fiksirana veličina i zavisnost heš funkcije od veličine heš tabele

# 10.5 - Jedan i dva nivoa direktorijima



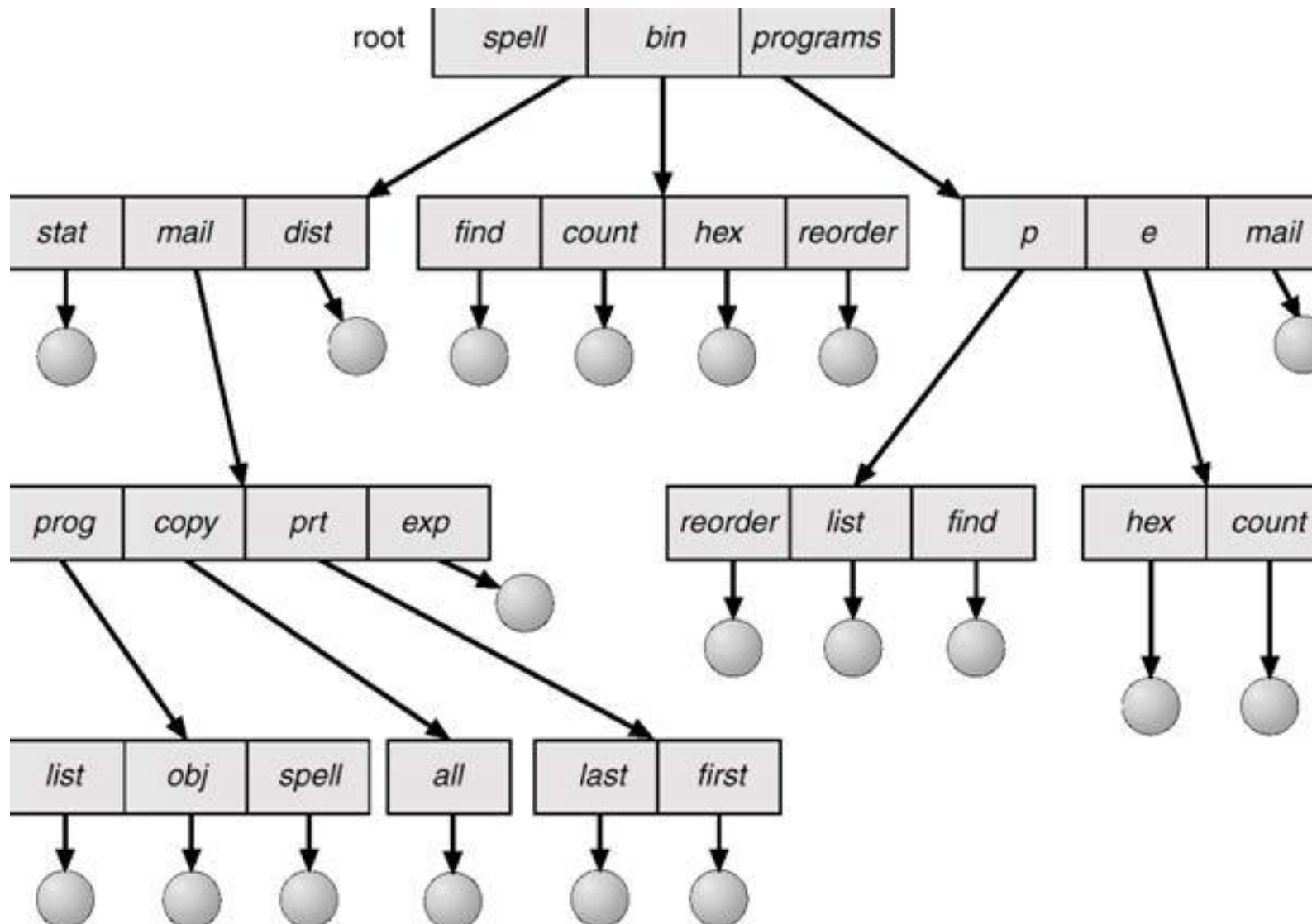
Jedan nivo



Dva nivoa



# 10.5 - Stablo direktorijima



Više nivoa

# 10.6 Dodela prostora - Kontinualna dodela

- Svaka datoteka zauzima **skup kontinualnih adresa** (prostor) na disku
- Za svaku datoteku u direktorijumu pamti se **početna adresa prvog bloka datoteke i broj blokova** koji su joj dodeljeni.
- Praćenje slobodnog prostora na disku - **kreira se lista slobodnog prostora**
- Lista sadrži **sve blokove na disku koji su slobodni**, tj. nisu dodeljeni
- Lista se implementira **kao bit mapa ili bit-vektor**.
- Svaki blok predstavljen je jednim bitom, **0-blok slobodan, 1-blok zauzet**
- Javlja se problem kada treba pronaći prostor za novu datoteku.
- Ovde se javlja problem **eksterne fragmentacije**
- Osnovni problem je odrediti **koliko prostora da se dodeli datoteci**.
- Problem **proširivanja datoteke** (dodavanje novih podataka)
- Datoteke koje narastaju sporo, tokom dugog vremenskog perioda u tom slučaju u dobroj meri **nepotrebno zauzimaju prostor**.

*Dobra strana ovog algoritma je jednostavnost i brzina.*

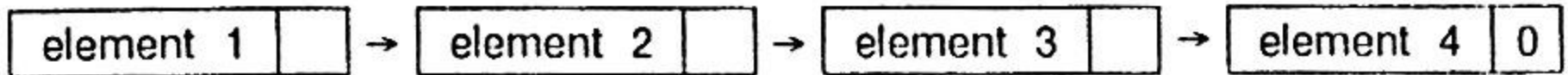
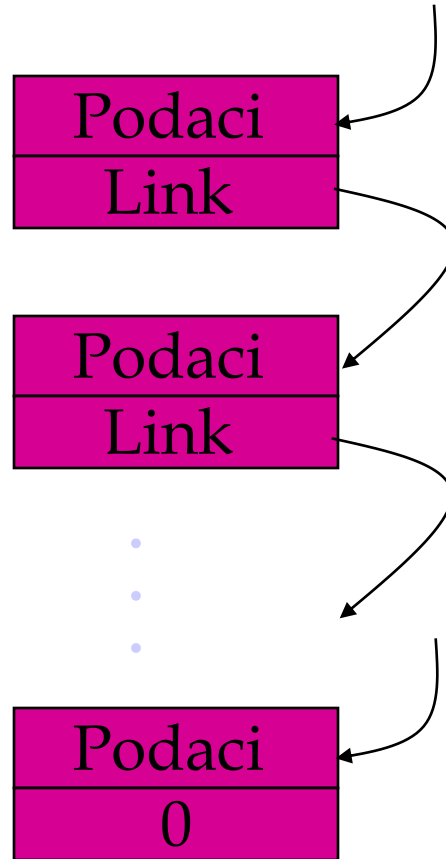
*Loša strana je ograničen prostor.*

# 10.6 Dodela prostora - Povezane liste

- Svaka datoteka predstavlja **povezanu listu blokova** na disku, pri čemu se ti blokovi mogu nalaziti **bilo gde na disku**.
- Direktorijum sadrži **ukazatelj** na prvi i poslednji blok datoteke.
- Koristi se metoda lančanih listi kod kojih **postoji ukazatelj u okviru samog bloka koji ukazuje** na sledeći blok
- Informacija o **slobodnim blokovima formira se kao posebna lista**
- Ova metoda je **mного sporija od mape bitova**, ali je količina sistemskih podataka o slobodnom prostoru na disku **značajno manja**.
- Ovde **ne postoji eksterna fragmentacija**.
- Pri kreiranju datoteke **nije potrebno deklarirati veličinu datoteke**
- Datoteka **može da raste** dokle god ima slobodnih blokova na disku.
- Ovaj metod **efikasan samo kod sekvencijalnog pristupa datoteci**.
- Za prelazak cele liste **treba utrošiti dosta vremena**.
- Mana je što se **troši prostor za ukazatelje** i smanjena je pouzdanost.
- Oštećenjem jednog bloka **gubimo pokazivač** na sve naredne blokove.

# 10.6 Dodela prostora - Povezane liste

Adresa prve liste



# 10.6 - Metod indekse dodele

- Ovde su **svi ukazatelji zajedno smešteni u indeksni blok**.
- Svaka datoteka **ima sopstveni indeksni blok** koji predstavlja polje adresa blokova na disku u kojima su podaci.
- Direktorijum **sadrži adresu indeksnog bloka**.
- Indeksna dodela **podržava direktni pristup**
- Eksterne fragmentacije **ovde nema**
- Postoji problem **utroška prostora zbog ukazatelja** u indeksnom bloku koji je veći nego kod povezane dodele.
- Ovaj problem je povezan sa veličinom indeksnog bloka pa se teži da on bude relativno mali, dok se **za veće datoteke ulančava više indeksnih blokova ili se koristi indeksiranje u više nivoa**.
- Još jedna alternativa je da u adresaru **čuvamo prvih nekoliko ukazatelja** indeksnog bloka, tako da za male datoteke indeksni blok nije potreban (UNIX operativni sistem).

# 10.7 – Realizacija sistema datoteka

Prilikom realizacije sistema datoteka, definiše se:

- 1. Logička struktura sistema datoteka** – podrazumeva na koji **način** je sistema datoteka predstavljen korisniku; pod ovim se podrazumeva definicija datoteka, direktorijuma, njihovih atributa i **dozvoljenih operacija** nad njima tj. sve što predstavlja logičku sliku sistema datoteka.
- 2. Fizička struktura sistema datoteka** – čine je **strukture podataka** na disku koje služe za skladištenje podataka. U ove strukture spadaju, **blokovi** (UNIX, Linux) i **klasteri** (*clusters* (*sistem datoteka FAT*))
- 3. Preslikavanje logičke strukture sistema datoteka u fizičku** – pod ovim preslikavanjem podrazumeva se uspostavljanje **veze između sadržaja konkretne datoteke ili direktorijuma i struktura na disku**; preslikavanje određuje da se sadržaj datoteke PRIMER.txt iz direktorijuma c:\1, nalazi u blokovima 15,16 i 25...

# 10.7 - Realizacija sistema datoteka

- Po pravilu, sistem datoteka se **realizuje u više nivoa** - slojeva (*layers*)
- Pri radu sa objektima sistema datoteka potrebno je **preći put od imena datoteke do konkretnog fizičkog bloka** na disku.
- **Nivo logičkog sistema datoteka** - na najvišem nivou, kome pristupaju korisnički programi, nalazi se logički sistem datoteka (*logical file system*). Na ovom nivou se **upravlja meta podacima**, tj svim strukturama podataka vezanim za datoteku osim njenog sadržaja.
- **Nivo organizovanja datoteka** - na sledećem nivou nalazi se modul za organizovanje datoteka (*file organization module*), koji vodi evidenciju o adresama logičkih blokova datoteka. Na osnovu zahteva za pristup delu datoteke, **modul određuje logičku adresu na kojoj se taj deo nalazi**
- **Nivo fizičkog sistema datoteka** - predstavlja osnovni, tj fizički sistem datoteka (*basic file system*), koji drajveru šalje osnovne komande za rad sa diskom. Ovde se **logičke adrese transformišu u fizičke adrese**.
- **Nivo upravljanja uređajima** - na najnižem nivou smešteni su drajveri za uređaj i rutine za obradu prekida. Upravlja se samim uređajima tj **podaci se prebacuju između diska i memorije**. Adresiranje je **fizičko**.

# 10.8 - Strukture podataka za pamćenje

Operativni sistem koristi **tri vrste struktura podataka** koje se primenjuju kod implementiranja datoteka:

1. **Sekvenca bitova** – ne vodi se računa o tome šta je u datoteci već je samo potreban podatak gde počinje **sekvenca za određenu datoteku i koliko je ona dugačka**. UNIX i WIN OS koriste ovu strukturu.
2. **Sekvenca slogova fiksne dužine** – imamo **fiksni broj bajtova udruženih u slog** koji se ređaju jedan za drugim. Nijedan operativni sistem opšte namene ne koristi ovu strukturu, već se ona **strogo koristi kod baza podataka** (*data base*) orijentisanim operativnim sistemima.
3. **Stablo** (indeksna sekvenca) - ovde su **čvorovi slogovi datoteke**. Svaki slog ima svoj ključ na fiksnoj poziciji u slogu. **Stablo se sortira po određenom ključu** i predstavlja vrlo efikasno rešenje kod velikih *mainframe* računara za brzi pristup datotekama.



# 10.8 - Strukture podataka za pamćenje

- 1. *BCB (boot control block)*** - BCB sadrži informacije koje su potrebne za otpočinjanje procesa podizanja jednog operativnog sistema. BCB je obično prvi blok sistema datoteka.
- 2. *Kontrolni blok particije (PCB, partition control block)***-sadrži podatke o sistemu datoteka, kao što su veličina bloka, ukupan broj blokova, broj slobodnih blokova sa pokazivačem na listu slobodnih blokova , listu slobodnih kontrolnih blokova datoteka i pokazivač na tu listu
- 3. *Kontrolne strukture za dodeljivanje datoteka*** - ovim strukturama se određuje konkretan sadržaj datoteke tj. logički blokovi u kojima je smešten sadržaj.U ove strukture spadaju tabela indeksnih čvorova (*inode table*)(*UNIX*), *FAT* tabela (*FAT*), *MFT*(*NTFS*).
- 4. *Direktorijumske strukture*** - sadrže kontrolne blokove datoteka
- 5. *Kontrolni blokovi datoteka (FCB, file control block)*** - sadrži attribute datoteka i opis prostornog rasporeda datoteke tj. pokazivače na blokove dodeljene datoteci.

# 10.9 - MS-DOS (FAT-12, FAT-16)

- Jednostavan sistem datoteka zasnovan na principu mape datoteka (realizovane u vidu FAT tabele sa 12 ili 16-bitnim adresiranjem).
- Ima ograničenje ukupne veličine sistema datoteka na 16535 klastera
- Veličina klastera praktično određuje i maksimalnu veličinu sistema
- Maksimalna veličina se postizala sa klasterima veličine 64 KB, što unosi značajne gubitke kroz internu fragmentaciju.
- Svaka datoteka u FAT ima svoje ime, datum, vreme i osnovne atribute.
- Ime datoteke se može zadati isključivo korišćenjem 7-bitnog ili 8-bitnog ASCII ili ANSI standarda.
- Pristupna prava i vlasnički odnosi nisu regulisani, tako da se na nivou sistema datoteka uopšte ne može kontrolisati pristup.
- Pouzdanost sistema je relativno niska: postoje dve identične FAT tabele koje su fizički smeštene jedna do druge na disku.
- U sistemu datoteka FAT, ne postoji dnevnik transakcija, tako da je upotreba *write-back* keširanja krajnje rizična.
- Sistem datoteka FAT danas se koristi pri formatiranju disketa.
- Pristup sistemu datoteka FAT može se ostvariti iz gotovo svih OS

# 10.9 - Windows - FAT32

- S porastom kapaciteta diskova, Microsoft je ponudio **FAT32** kao proširenje sistema datoteka **FAT**.
- Osnovna promena u odnosu na stari FAT **predstavlja 32-bitno adresiranje**, čime je omogućeno formiranje većih sistema datoteka s klasterima prihvatljive veličine.
- Omogućeno je formiranje **velikih particija većih od 2 GB** na diskovima većim od 8 GB koji su se tada pojavili.

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

# NTFS (New Technology File System)

- Ovaj fajl sistem koristi **64 bit-ne adrese** - veličina particija **do  $2^{64}$  bajta**
- Korišćenje NTFS tehnologije znači veću **sigurnost**, tj. zaštitu i veću **toleranciju i korekciju greški**.
- **Potpuni pristup** podacima na sistemu datoteka NTFS isključivo je moguće iz operativnih sistema Windows 2000/XP/2003 i novijim.
- NTFS se može aktivirati u **režimu čitanja i pisanja i pod UNIX/Linux**
- Maksimalna veličina NTFS volumena je **od 2 do 16 TB**
- Maksimalna veličina datoteke **određena je veličinom volumena**
- NTFS sistemi datoteka **ne mogu se formirati na disketama**
- Osnovna struktura sistema datoteka NTFS jeste **volumen zasnovan na logičkoj particiji diska**.
- Volumen može **zauzeti deo diska, ceo disk ili više diskova**.
- **Svi metapodaci**, kao što su informacije o volumenu, **smešteni su u regularnoj datoteci**.
- NTFS **koristi klasterne kao osnovne jedinice dodele diska**.
- Klasteri se na NTFS sistemu adresiraju korišćenjem logičkih brojeva klastera (*Logical cluster numbers, LCN*) **kao adresa diska**.

# NTFS (New Technology File System)

## Prednosti NTFS-a u odnosu na FAT i FAT32:

- **Otpornost u slučaju otkaza** (*fault tolerance*) postiže se korišćenjem RAID tehnike.
- Ukoliko OS **otkrije neispravan deo na disku**, iskoristiće ispravnu kopiju podataka i **napraviti nov sektor** koji će nadalje koristiti
- Mogućnost pravljenja **aktivnog direktorijuma** (hijerarhijske baze distribuirane na kontrolerima domena)
- Mogućnost **kompresije datoteka i direktorijuma** u cilju smanjenja zauzetog prostora na diskovima
- Mogućnost **šifrovanja datoteka** radi povećanja sigurnosti
- Kontrola pristupa **pomoću određenih skupova pristupnih prava** dodeljenih ovlašćenim korisnicima i grupama
- Praćenje **aktivnosti na sistemu datoteka**, koje omogućava **brz oporavak od logičkih oštećenja** (dnevnik transakcija)
- **Disk kvote**-kontrolirše se zauzetost sistema datoteka od strane korisnika kojima se može na upotrebu dodeliti određeni prostor na disku
- **Prelomnetačke** koje omogućavaju izvršavanje različitih programa

Hvala na pažnji !!!



Pitanja

? ? ?